AD-A096 544   ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB   F/G 5/2
THE STATE-OF-THE-ART IN NATURAL LANGUAGE UNDERSTANDING.(U)
JAN 81   D L WALTZ   N00014-75-C-0612
UNCLASSIFIED   WP-27   NL

1 OF 1
AD A
096544

END
DATE
FILMED
4-81
DTIC

6 The State-of-the-Art in Natural Language Understanding.

9 Working Paper 27

10 David L. Waltz

Advanced Automation Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

11 28 Jan~~uary 20,~~ 1981

14 WP-27

Abstract

Research in computer understanding of natural language has led to the construction of programs which can handle a number of different types of language, including questions about the contents of data bases, stories and news articles, dialogues, and scene descriptions. This research draws on and has in turn had an effect on many other research areas, including software engineering, linguistics, psychology, philosophy, and knowledge representation. This paper provides a brief history and overview of the field, along with examples and explanations of the operation of several natural language understanding programs. The limitations of our current technology are discussed, and assessments are given of the most promising current research directions.

12 35

key words and phrases

"Natural language, natural language understanding, natural language processing, computational linguistics, natural language systems, history of natural language research, artificial intelligence."

15 N00014-75-C-0612

This work is supported by the Office of Naval Research under Contract N00014-75-C-0612.

Working papers are informal papers intended for internal use.

1. Introduction

The original purpose of this paper was to give some answers to the following questions about the state-of-the-art in natural language understanding systems:

What are the limits now?

What are the obstacles to progress?

Where are the answers likely to lie?

In order to be able to answer these questions, I first set out what I feel are the major lines of natural language research today, including the study of such topics as knowledge representation, metaphor, "speech acts" (the use of language to achieve goals), modeling of "common sense" and plausibility judgement, relationships between language and perception, etc. One way to make sense of this potpourri of research topics is to consider the basic questions being explored by two or more of the research areas. Looked at this way, I think that the following five questions are motivating much of the current research in natural language understanding:

(1) What is the function/purpose of language?

Language is in general used by a speaker to achieve goals. Unless we understand these goals, we cannot understand the language. Goals may be extremely complex: a speaker may mean to inform, correct, or mislead a listener; or a speaker may wish to have the listener perform a physical or cognitive action, or undergo a certain kind of experience, or answer questions, and so on. Often it is necessary to have a model of the speaker's ordinary behavior in order to understand the speaker's goals -- the language alone may not be sufficient. And in order to tell whether or not a speaker is telling the truth, a listener must be able to compare the speaker's language with models embodying knowledge of human behavior as well as the behavior of the physical world. To make matters even more problematic, any given utterance may be used to serve quite different goals in different situations, and a single utterance may serve multiple goals simultaneously.

(2) What does it mean to "understand language" and how could we show that a system can understand?

Research attention has focused on the sentence for a long time. However, many important units of language are much larger than the sentence: dialogues, instructions, scene and event descriptions, stories, explanations, etc. We currently lack the ability to assign meaning and purpose to all but the very simplest of these larger units of language.

(3) How can a program deal with novel language?

The preponderance of work to date has allowed us to deal with novel syntactic structures, but we have relati.ely very little understanding of methods for dealing with novel semantic structures, and virtually no methods for dealing with novel concepts expressed in language. We have a need for semantic methods which can give us meanings for phrases, e.g. "engine housing acid corrosion damage report summary"; we also need a dramatically expanded understanding of metaphor and other non-literal lanugage (e.g. "The soldiers were shattered by the experience" or "We found a refrigerator graveyard.").

(4) How can a program judge whether language is meaningful?

How do we know that "The man jumped over the fence" can be literally meaningful whereas "The cow jumped over the moon" can not? How can we decide that a message is garbled or that its sender is deranged? How can we decide that a metaphorical interpretation is intended, and how can we know that a given metaphorical interpretation is sensible? To answer such questions a system needs "common sense", and common sense must surely be based on an extensive and detailed model of the physical world, as well as of the worlds of human action and inner experience (e.g. perception, emotion, memory, etc.).

(5) What is the most effective way to make the restricted natural language systems of the foreseeable future seem natural to humans?

We have only the beginnings of an understanding of how users will behave with natural language systems. Thus there has been to date a fair degree of mismatch between systems and users. We would like to be able to evaluate both existing systems and future design alternatives for usefulness and convenience. We would like to able give casual users systems that allow natural expression, that do not often surprise users by not understanding or misunderstanding their language.

The rest of this paper is organized historically. I could not find a good way to fit together the five questions above to form a coherent picture of the current state-of-the-art of research, and I found it was even more difficult to show how the current questions related to the ultimate natural language processing questions. I discovered, however, that current research directions seemed much more sensible if they were viewed as responses to specific shortcomings of earlier ways of looking at the process of natural language understanding.

2. Ancient History

Before 1940 computers, if they were thought about at all, were considered to be number processors. During the 40's, two major developments led to the view of computers as somewhat more than simple number processors. The first set of ideas was due to McCullough and Pitts, who theorized that each neuron is a logical device (roughly an AND or OR gate). We now know that each neuron is far more complex than they believed it to be, but their ideas were important in that they suggested that all intelligent processing, whether arithmetic or symbolic, numerical or verbal, could be performed by a single type of mechanism. Thus their views were important in a much more precise formulation of the brain-computer analogy than had been possible before.

The second major piece of work was Shannon's work on information theory; Shannon showed that both numbers and text could be treated as special cases of a more general concept he called "information", that information content could be quantified, and that ideas about information had interesting mathematical and practical applications.

2.1 Machine Translation

Shannon's work led in the early 50's to what I will call "the era of machine translation". Being able to treat text and language in general as information allowed the possibility that language might be manipulated on the new digital computers that were then being constructed. The initial idea for machine translation was the following: translation is a process of dictionary look-up, plus substitution, plus grammatical re-ordering. As an example, the English sentence, "I must go home" could be translated into the German "Ich muss nach Hause gehen" by substituting "Ich" for "I", "muss" for "must", "gehen" for "go" and "nach Hause" for "home". In the process two words, "nach Hause" (to the house) had to be substituted for "home" -- we won't worry here about that fine point -- and a simple kind of grammatical re-ordering had to take place to move the verb to the end of the sentence.

For simple examples this model of the possibility of translation seems rather intriguing. However, it soon became clear that translation is really not possible without understanding. To illustrate the need for understanding in translation, a classic story (probably apochryphal) describes the machine translation of the phrase "The spirit is willing but the flesh is weak" into Russian and then back into English; the translation is said to have come out: "The vodka is strong but the meat is rotten."

Clearly a greater amount of world knowledge was needed; a program had to understand what was being said in order to be able to translate it properly. Yet another classic example was given by Bar-Hillel in a 1964 paper in which he explained why he was leaving the field of machine translation. Bar-Hillel cited the sentences, "The pen is in the box" and "The box is in the pen", and pessimistically stated that he could not imagine how a machine could translate both sentences correctly, assigning "pen" the meaning "writing implement" in the first sentence, and "playpen" or "stockpen" in the second. While we still have a long way to go before we could claim to have programs that truly understand or translate a significant range of types of language, we do now know how to write programs that can appropriately assign different meanings to "pen" in Bar-Hillel's examples above by using a system which can manipulate simple spatial models of objects [Waltz 80].

The work on machine translation did give a great deal of impetus to work on syntactic theory as evidenced especially by the work of Chomsky and also to a degree in the early work on parsing high-level languages for compiler construction, now a core topic in computer science.

To continue this brief history, other major ideas that have been influential in the history of natural language processing surfaced in the 50's. I refer specifically to the introduction of the idea of heuristic search by Newell and Simon in 1955 and also to the introduction of the LISP programming language by McCarthy in 1959. Most natural language processing systems have been written in LISP.

The entire field field of machine translation essentially came to an end in the early 60's. It is only now undergoing a kind of renaissance, using AI models of meaning, but the early effort was a nearly complete failure.

3. The Semantic Information Processing Era

Out of the rubble of machine translation effort grew an effort that is closely associated with artificial intelligence. The "semantic information processing era" (roughly 1962-1973) produced a number of ideas used in today's natural language application systems, some of which have proved to be of practical value. Some notable ideas of this era are the following:

(1) the use of limited domains for language understanding systems: rather than attempting to understand all language, the limited domain approach is to design a system that is expert in one specific area of language, but perhaps knows nothing at all about any other domain;

(2) the "big switch" theory -- to rationalize the study of limited domains as a contribution to a full cognitive theory, the "big switch" theory was advanced; the big switch theory holds that it is possible to construct a broadly intelligent system by generating experts in a number of limited domains and then piecing together a huge system containing these experts along with a special expert, the "big switch", which could select the appropriate expert to handle any given problem;

(3) the use of key words to trigger certain actions -- natural language programs using this idea look in a sentence for one or more key words and, on the basis of what is found, take appropriate action (I give an example below);

(4) the "translation" of English into formal languages -- some of the formal languages that have been used include predicate calculus, data base query languages, and sets of linear equations.

Overall, we could characterize the approaches of the 60's to natural language processing as "engineering approaches", approaches which attempted to solve specific problem domains, not to embody psychological reality. What do I mean by "engineering approaches"? Let us look at some examples.

3.1 Keyword Systems

The first example is the use of key words. Key words were particularly important in the ELIZA and DOCTOR programs written by Weizenbaum [1966], and the PARRY program (which simulated a paranoid person) by Colby and his collaborators [1975].

PATTERN	RESPONSE
(* computers *)	Do computers frighten you?
(* mother *)	Tell me more about your family.
<nothing matches>	Please go on.

Figure 1. Simplified ELIZA patterns and responses.

In Figure 1 (a highly simplified example based on ELIZA) "*" matches any word or list of words (including no words at all) and the literal words such as "computers" can only match words like "computers". Thus if someone were to type "I hate computers" to the ELIZA program, it might respond, "Do computers frighten you?" If the person typed, "My mother is an electrician," ELIZA could respond, "Tell me more about your family". ELIZA was also capable of using phrases and words which matched patterns to construct responses; thus, it could respond to "I believe that <x>" with "How long have you believed that <x>".

3.2 Translating English into a Formal System

As an example of the translation of English into a formal language, consider Bobrow's STUDENT program [Bobrow 1968] which translated algebra word problems into a set of linear equations. STUDENT treated each input sentence as though it corresponded to a simple equation; thus, "John's age now is two times Mary's age" would be translated into an equation such as "JA = 2 * MA". In order to perform this translation, Bobrow's program had to note that John's age now is a variable (JA), Mary's age is a variable (MA), and "is two times" should be translated into "2 *" in the equation. Similarly the equation, "In three years John will be six years older than Mary" translates into the equation "JA + 3 = MA + 6". This program, once it had formed as many equations as variables, could then pass the equations to another program that was expert at solving simultaneous linear equations. The idea of translating English into formal languages has led to many other programs including most of the current generation natural language data base "front ends".

3.3 Data Base Question-Answering

Another precursor of data base query generation from English was the BASEBALL program of Green[1963]. BASEBALL had a tabular data base much like that shown in Figure 2a, containing information about all the games played in the American League during one season. When given a question such as "Who did the Yankees play on July 7?", the BASEBALL program turned this into a query template similar to the one shown in Figure 2b. BASEBALL could then compare this query template with the data base and return the answer "Red Sox".

MONTH	PLACE	DAY	GAME	WINNER/SCORE	LOSER/SCORE
July	Cleveland	6	95	White Sox/ 2	Indians/ 0
July	Boston	7	96	Red Sox/ 5	Yankees/ 3
July	Detroit	7	97	Tigers/ 10	Athletics/ 2

Figure 2a. BASEBALL's data base.

```
(OR (July  7  --  Yankees/ --   ?ANSWER/ --)
    (July  7  --  ?ANSWER/ --   Yankees/ --))
```

Figure 2b. A query template in BASEBALL.

All these programs illustrate some of the kinds of "engineering techniques" that were used to handle language during the 60s, techniques which illustrated the simplifications possible through the restriction of inputs to narrow semantic domains, and which offerred the promise of near-term practical applications. Unfortunately, the programs developed using these techniques shed very little light on the cognitive processes underlying language comprehension.

4. 1970 - The Flowering of Semantic Information Processing and Seeds of Cognitive Science

The years around 1970 proved to be noteworthy for a number of reasons. I will describe briefly several well-known and influential programs that appeared around 1970, and which pushed the notion of semantic information processing to its ultimate limits.

4.1 SHRDLU

The first program is Winograd's SHRDLU [Winograd 1972]. Winograd's program assumed that two main analogies were true. The first was that sentences were analogous to programs, that is, that sentences could be "understood" by transforming them into programs. The programs thus created could then be used to carry out various tasks (e.g. moving blocks on a table), or search for information in SHRDLU's data base, or generate an answer for its user. The second, related analogy was that words correspond to program steps. Thus, word "definitions" for SHRDLU were program fragments in the MICROPLANNER programming language [Sussman and McDermott 1972]; MICROPLANNER, inspired by Hewitt's [1969] PLANNER language, was centered around the ideas of heuristic search. Thus, Winograd's program unified the much earlier heuristic search material with natural language processing. SHRDLU's grammar itself operated by a heuristic search process; it tried out a certain interpretation of a sentence, and if it could not make syntactic or semantic sense of the sentence, SHRDLU would back up and try a different interpretation.

SHRDLU solved a broad set of problems, and was able to handle sentences exhibiting a wide variety of linguistic phenomena. It interpreted declarative sentences as data base updates, interrogative sentences as data base searches, and imperative sentences as specifications for goals; these goals were achieved by first forming and then executing a plan, which generally involved data base search and update as well as sentence generation. Winograd's dissertation also included detailed analyses of SHRDLU's limitations, and listed a number of examples that would be difficult to handle within Winograd's paradigm. SHRDLU's linguistic coverage was very broad compared to previous programs. It was able to handle, for instance, quantification, some kinds of pronouns, and negation. It was able to learn word definitions, generate natural-sounding dialogue, and answer questions about the history of its dialogue and plan execution. An example of the operation of Winograd's program is shown in Figure 3.

Find a block which is taller than the one you are holding and put it into the box.

By "it", I assume you mean the block which is taller than the one I am holding.
OK.

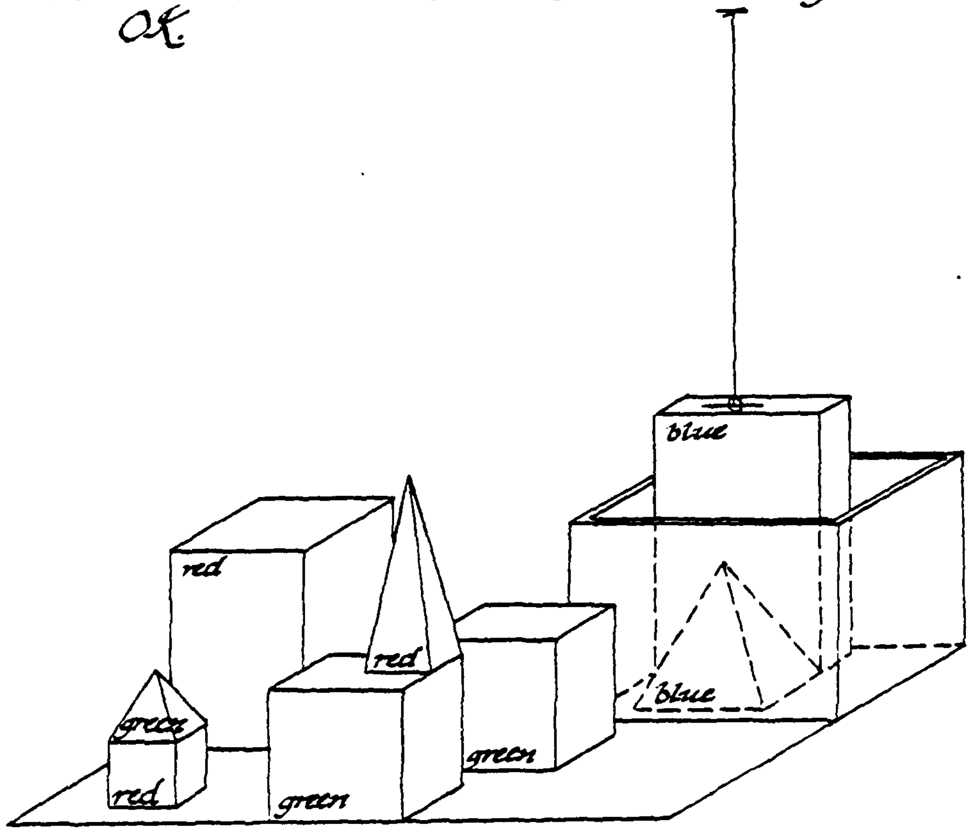

Figure 3

4.2 LUNAR

The second piece of work from around 1970 which I would like to discuss is Woods' LUNAR program [Woods et al 1972], which was a natural language front end for a data base containing moon rock sample analyses. For parsing sentences (i.e. finding the syntactic structure of the sentences) Woods used Augmented Transition Networks (ATNs) [Woods 1970], which implemented a heuristic search much like the kind that Winograd used in SHRDLU. Woods' formulation was so clean and natural that it has been used since then for most parsing and language understanding systems. Woods also introduced a very general notion of quantification based on predicate calculus and used sophisticated techniques to translate questions into data base queries. An example of a sentence that Woods' LUNAR program could answer is: "Give me all analyses of samples containing olivine."

Both LUNAR and SHRDLU were comprehensive systems; both could use relatively unconstrained language; both worked in very narrow domains, but had complete, privileged knowledge of their worlds. (LUNAR knew everything that could be known about the data base of lunar rocks; SHRDLU was the keeper of the block's world.) Both also have proved to be non-portable and non-extensible. Although there were several attempts, no serious production programs ever developed from either of the pieces of work. Both were prototypes which had a limited life and are now no longer used.

4.3 NLPQ

NLPQ, a third interesting program from around 1970, came out of the work of George Heidorn on the use of natural language to set up simulations [Heidorn 1974]. For example, given the following sentences (a partial transcript of the program's operation) Heidorn's program could set up a simulation, and run it to answer questions:

User: When a vehicle arrives at a station, it leaves there immediately if the length of the line at a pump in the station is not less than 2.
75 percent of the vehicles are cars and a fourth are trucks.
There is just one pump.
A simulation time of 8 hours is desired.
Ask questions for further info.
System: HOW OFTEN DO THE VEHICLES ARRIVE AT THE STATION?
User: The arrivals of vehicles are normally distributed with a mean of 8 minutes.

<system asks more questions, eventually judges that the problem statement is complete, and can then answer questions about the situation that was described to it or about the simulation itself>

Heidorn's program embodied a model of what a complete simulation would have to include and was able to ask questions of the user if the information that was given was insufficient. Thus, Heidorn's program embodied a kind of world knowledge about what constitutes a complete formulation of a problem.

4.4 MARGIE

Also around 1970 another influential piece of work by Roger Schank was completed. This work has continued to this day. Schank has dealt with much more unconstrained language, particularly language about human actions. Schank's work was based on the development of a set of "primitives of conceptual dependency". All sentences input to Schank's systems are translated into structures centered around a small number of primitives. The primitives (which have changed a little over the years, and have varied in number from 14 to 16) include MTRANS, which stands for "transfer of mental information"; ATRANS, which stands for "transfer of possession"; PTRANS, which stands for "physical transfer of an object from one location to another"; CONC, short for "conceptualize", or think about; MBUILD, which stands for "build memory structures"; ATTEND, which covers see, hear, taste, smell, touch; PROPEL, which stands for "the application of physical force to an object"; MOVE, that is, move a body part; GRASP, that is hold in one's hand; INGEST, and EXPEL. Sentence meaning representations are formed by using these primitives in conjunction with other words in a sentence to form a kind of "semantic network" (see examples below). Each primitive of conceptual dependency is also associated with a case grammar-like frame [Fillmore 1968] that specifies which words can occur with the primitive in a sensible manner. Thus, for instance, MTRANS (the transfer of mental information) requires that there be an intelligent source for the mental information and an intelligent recipient for the information. (MTRANS is the primitive used internally to represent such diverse words as tell, hear, say, speak, read, etc.)

Conceptual dependency primitives have been used not only to represent meaning but also to organize expectations; for example, having decided that mental information was transferred to a hearer Schank's programs could predict that the hearer would thereafter have that information available in memory.

The MARGIE program [Schank et al 1973] could accept simple sentences and answer questions about them, generate paraphrases of those questions, and make inferences based on the questions. For example, given the statement, "John gave Mary an aspirin," the inference program generated sentences such as: "Mary felt sick," "Mary wanted to feel better," "John Wanted Mary to feel better," "Mary asked John for an aspirin," etc. (All these were viewed as plausible, not necessary inferences.)

Figure 4 shows some examples of conceptual dependency diagrams corresponding to input sentences. In Figure 4a is a structure corresponding to "John grew six inches". One can read this roughly as "John's size went from some value X to some value X + 6 inches". The representation of the apparently similar sentence, "John grew corn" is quite different, as shown in Figure 4b. This structure can be roughly read, "John did something (unspecified) which caused the size of the corn to go from some size X to some size X + Delta".

Figure 4c is a conceptual dependency diagram corresponding to a sentence, "John gave Mary a bicycle." This structure can be roughly read, "John transferred possession of the bicycle from himself to Mary."

In Figure 4d the related sentence, "Mary got a bicycle from John" has a very similar representation except that Mary is listed as the agent, i.e. the actor who caused the transfer of the bicycle from John to her.

4.5 Other Ideas from the early 70's

Also in the early 70's there were several other contributions that have played an important role in defining current research topics. Two such contributions were made by Searle [1970] and Grice [1975] on "speech act theory".

Speech act theory attempts to account for the purposes for which language is used, as opposed to the (logical) meaning of individual sentences. As an example, if given the sentence, "Could you pass the salt," we understand this not as a request for information about whether we are physically capable of passing the salt, but as a request to carry out the action of actually doing so. In this sense, speech act theory points out that sentences are not analogous to programs -- that is, that no direct translation of a sentence into a program form will capture all its meaning(s). Language consists of acts by speakers, and as such, the intentions, goals, strategies, and beliefs of both speakers and listeners are of central importance in understanding language.

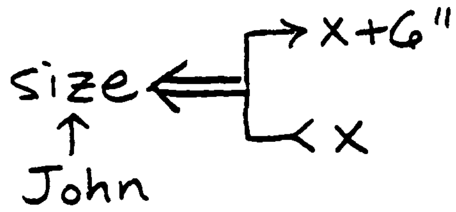

Figure 4a.
"John grew six inches."

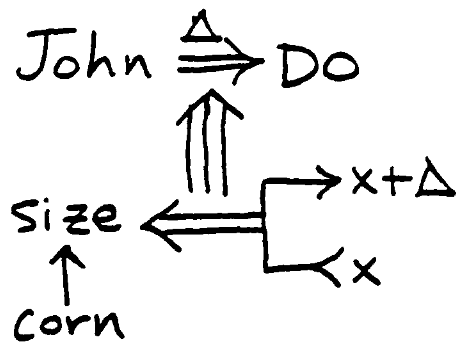

Figure 4b.
"John grew corn."

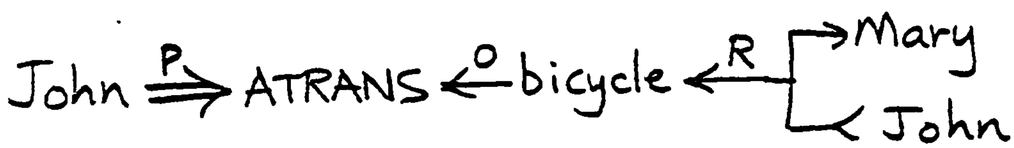

Figure 4c.
"John gave Mary a bicycle."

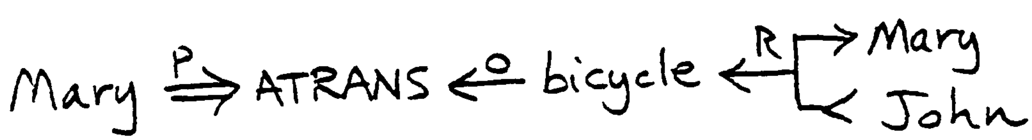

Figure 4d.
"Mary got a bicycle from John."

5. Lessons of the 70's

Other phenomena were noted during the 70's that we still do not know how to deal with well. These include processing language which falls outside of a narrow domain, handling dialogue (which can itself be the topic of conversation in any dialogue), insuring rapid response, and meeting other "human factors" requirements (such as presenting information to a user of the natural language system in a way that is unambiguous, teaches the user about the system's abilities and limitations, etc.). Speed is especially important. If users have to wait a long time for a response, they can become very impatient, so efficiency of the algorithms for natural language processing is of importance. (I'll have more to say about some of these issues on dealing with real users when I talk about the accomplishments of the PLANES system.)

5.1 Knowledge Representation

Another major realization during the 70's has been that knowledge representation formalisms are of central importance to all natural language processing. Before we can put knowledge into a system, we need to be able to represent that knowledge appropriately, in a manner which allows the knowledge to be found and used when appropriate during the natural language understanding process. This need has been pointed out for a long time by John McCarthy [1968] and is now generally recognized as the central issue in artificial intelligence. Among the issues in knowledge representation are: how should items in memory should be indexed and accessed, how should context be represented, how should memory be updated, how can programs deal with inconsistency -- that is, if we have new information to be added to our knowledge base which is inconsistent with the information currently there, how should we store the new information? can we (and should we) resolve the conflict? If not, which information should we act on, or should we somehow integrate both parts of the conflicting information into our action? Various ways have been suggested for handling this sort of conflict, for example, partitioning memory into a number of possible "contexts", each of which is internally consistent. Another important problem is that of deciding whether and how we could know that a knowledge representation scheme is sufficient and complete, so that we could be assured that any kind of knowledge imaginable could be represented in the scheme.

5.2 Common Sense

We came to realize during the 70's that we needed to endow natural language programs with "common sense", which can only be based upon a body of knowledge of the outside world. In understanding language, people bring a large amount of information to bear which cannot be deduced from the language itself. A sentence is never a formula or a program which is complete in and of itself. No process of rewriting a sentence could be sufficient to construct all the meanings that a hearer gets from listening to the sentence. In many ways, language is a kind of shorthand or set of index items; a listener uses these as keys, and retrieves from memory the rest of (appropriate) information that must be added to the language in order to formulate its full meaning.

Consider, for example, the following sentences (from [Winograd 1972]):

The city councilmen refused to give the women a permit ot march because
(a) they feared violence.
(b) they advocated revolution.

In (a) they seems to refer to the city councilmen, whereas in (b) they refers to the women. How do we judge this? The structures of the two sentences are identical, so they cannot help us to distinguish the two cases. The only answer seems to be that we know a great deal about human behavior, and can readily access and apply this knowledge when it is needed in understanding language.

One problem requiring common sense is the problem of judging whether or not a sentence is even meaningful. Related problems involve choosing the most appropriate reading when several are possible, and making appropriate inferences about sentences or text passages. Some research from the mid-70's gave some tentative answers to these kinds of problems.

5.3 Frames

In his "frames theory" Minsky [1975] suggested that we needed to be able to deal with much larger memory units than had been considered before. He offerred as candidates for the memory units "frames", structures consisting of a core and slots: each slot corresponding to either a facet or participant of a concept embodied in the frame, or a space for a pointer to a related concept e.g. an instance of the frame's concept or a variation on the frame). Minsky argued that an important function of frames was to represent stereotypes; stereotypes provide a neat explanation for "default reasoning", the process by which we take the shorthand information available in language, and retrieve and fill in the rest of the information that would ordinarily be be expected in that situation.

Frames were also suggested for modeling context; that is, a context could be represented as a frame which in turn would contain as slot values other frames that ought to be present in that context.

5.4 SCRIPTS

Another larger processing unit specialized for stories is the SCRIPT, proposed and developed by Roger Schank and his collaborators at Yale [Schank and Abelson 1977]. SCRIPTs correspond to stereotypes for stories, and are proposed as the kind of information that allow us as listeners of a story, to fill in unmentioned details and make appropriate inferences. SCRIPTs also can also provide a plausible mechanism for expectation-driven text analysis. If we know a story is about a restaurant, we expect that we may encounter a waitress, menu, table, a bill, food, and other specific kinds of information; SCRIPTs provide a kind of ready-made framework for encoding that kind of information. As an example of the use of a script, consider the following story (a simplified version of a story actually used by Wendy Lehnert [1977]).

> John took the bus from New Haven to New York. On the way, his pocket was picked. He went to Mama Leone's and ordered spaghetti. John couldn't pay the bill, so he washed dishes.
>
> What did John eat?

Notice that in the passage it was never mentioned that John ate spaghetti. It says that he ordered spaghetti, yet we make the inference that he actually ate the spaghetti in the absence of any information to the contrary. Lehnert's program used a restaurant SCRIPT to make plausible inferences.

5.5 Non-literal Languge

Another realization of the 70's was that we needed different or new techniques for dealing with non-literal language. As pointed out by a number of workers, metaphor is a pervasive phenomenon in language. Typically words have many senses which are not neatly captured by a simple definition. Words can be applied in novel situations which are difficult to predict from any number of dictionary definitions. Some attempts to deal with nonliteral language included the "preference semantics" of Wilks [1976], and Becker's [1975] "phrasal lexicon", a compendium of idioms which cannot be understood as a composition of simple word definitions. Examples include "big as a barn", "sly as a fox", "dry as a bone", and so on.

It was also recognized that new techniques were needed for dealing with language units that were larger than sentences. Such instances included stories or news articles, dialogues, and descriptions or instructions.

5.6 Evaluation and Data on Users

One of the difficulties in evaluating the current state of the art in natural language processing is that most papers only give positive examples of the operational systems. There is no way to tell whether these positive examples are typical of the operation of the system, or whether they are an exhaustive list of all the questions the system has ever answered appropriately. In addition, we have little information on how users will behave with a natural language system if they are not constrained. We do have some experience with tests where a human simulates a computer [Malhotra 1975][Tennant 1980]. In such tests users sit at a terminal and type into it as though they were typing to a natural language understanding program, when in fact they are typing to another terminal where a person is sitting pretending to be a natural language understanding program. Such tests are probably too unconstrained because the person simulating the natural language system is obviously capable of understanding all sorts of language. Nonetheless, such tests are very instructive in gaining an understanding of how users would behave with an ultimate natural language system. Such tests give much less information about what the minimum features should be included in order to make a useful natural language system. Some of the questions we would like to know about user behavior are the following: Which features should a system have? Which <u>must</u> it have? Which features are the most important? What computational model most naturally handle such features? And finally, is it possible to have a restricted natural language system that would be truly convenient for a casual user?

6. Natural Language Front Ends for Data Bases

During the 70's a number of natural language data base front ends appeared: LUNAR [Woods et al 1972] has already been briefly described; other systems included REL [Thompson et al 1969, 1975], an English-like extensible system; LIFER/LADDER [Hendrix et al 1978]; REQUEST [Plath 1976]; and ROBOT [Harris 1977].

6.1 PLANES

As an example of the engineering approach and to give a more complete idea of the current state of the art of natural language processing, I would like in this section to discuss the PLANES system developed at the University of Illinois. PLANES is a natural language data base front end that works on a large relational data base of aircraft flight and maintenance data. PLANES assumes that all the language it obtains is in the form of requests which it turns into formal query language expressions. It then runs the query language expressions on the data base and returns an answer to a user in an English-like or tabular form. PLANES uses a "semantic grammar", that is, it has ATN parsers for every kind of phrase that can occur in its world: time phrases, phrases referring to aircraft, places, etc. The goal of the parsing phase of PLANES is a set of semantic constituents; so for example, the sentence, "Which planes had 10 or more flights during January 1970?" yields the semantic constituents "Which plane"; "greater than ten flights"; and "January 1970". The goal of the front end is to take these constituents and fit them into a "query template". The query template may not be filled in completely by the information that was given in the English sentence and in this case, PLANES looks back through the dialogue to locate and fill in missing items. Thus, given the pair of requests:

Which aircraft required more than 10 hours maintenance in June 1979?

<answers question>

July?

PLANES would use the information from the first sentence in formulating a query expression for the second sentence.

PLANES was designed from an engineering point of view and makes no pretense of modeling psychological reality. Some of the advantages of the design of planes are the following: (1) It allows a nongrammatical input. The phrases can occur in any order so that, for instance, the sentence, "A7s January 1970 unscheduled maintenance" is a reasonable request for PLANES. (2) PLANES can handle ellipsis, as illustated in the example above; (3) PLANES can also handle some forms of pronoun reference in a manner similar to the way it handles ellipsis. (4) It can deal with unforeseen requests because it is able to use the query template to do expectation-driven analysis of such requests. It fills in the query template by categorizing all the constituents it finds and then inserting those constituents in the appropriate slots in the query template.

PLANES handles speech acts by matching all the speech act words it can find and then ignoring them; PLANES always assumes that the user is requesting information. Thus PLANES matches and then throws away all such portions of requests as "Can I get ...", "Could I get ...", "Can I have ...", "Could you give me ... ", "Could you show ...", "Could you get ..." and so on.

PLANES judges the plausibility or meaningfulness of questions through reference to what we call "concept case frames". Some examples of concept case frames are [<plane><receive><maintenance>] or [<plane><fly><flight hours>]. The former can match sentences such as "Which planes had maintenance in January?" or "Did any planes have unscheduled maintenance during January?" and the latter can match sentences such as "How many flight hours did plane 7 log in February?" Concept case frames are also used for ellipsis and pronoun reference. If some constituents are missing PLANES uses concept case frames to decide which constituents it needs to locate in the preceding dialogue in order to form a complete request.

Some aspects of plausibility judgement are also embodied in the semantic grammar since PLANES parses time phrases, place phrases, airplane phrases, and maintenance type phrases separately. It can make judgements as to whether each of these individual phrases is meaningful.

6.2 User Evaluation of PLANES

In testing, we have found that users often ask vague and complex questions. For example, users ask for reports such as in the following: "Give me a month by month status report for F-4's." The system is simply incapable of deciding easily what "status report" means in this example. In other cases, the system may be asked to make judgements as in the sentence, "Which plane had the worst maintenance record?" The problem here is that worst maintenance record can mean different things to different users. Is the worst plane the one that had the most hours out of service, or is it the one that cost the most to repair, or is it the plane that required the most maintenance hours during the month, or is it the class of plane that crashed most often that month? Without knowing more about the user and his interests, we simply cannot make such a judgement; as it stands, PLANES cannot even be programmed to include representations of these various possibilities.

Users often input declarative information. For example users may say to the system, "I'm only interested in A-7's". While it can deal with certain special cases of this sort, PLANES is incapable of dealing with such sentences in an appropriate way in general; it assumes that user inputs are requests, unless it recognizes them specifically.

Users often refer to items which are not in the data base. For example, a user can refer to concepts from earlier sentences or answers to previous questions, and PLANES is not currently capable of dealing with such questions. In addition, users sometimes refer to items that are not covered within the data base scope. For instance, our data base has no information on pilots or the sources and destinations for specific flights. Yet a user might reasonably be interested in asking such information and PLANES would simply say, "I did not understand your request."

Rewriting PLANES for a new data base would be difficult. Much of the information in PLANES, including the semantic grammar for handling plane phrases and time phrases, would not carry over to a different world, and a new semantic grammar would have to be generated.

PLANES does not respond well when it only partially understands the question. Sometimes in such cases PLANES will simply say, "I did not understand your question." If no answer is found, PLANES simply says, "I found no answer." It could be that no answer was found because there was no data of the sort the user was looking for, and PLANES would not easily allow a user to distinguish this case from the one where there were no instances of the specific type of event the user was looking for. This latter ability is being added to PLANES.

6.3 Evaluation with Casual Users

In recent testing [Tennant 1980], users were briefly (in less than 15 minutes) introduced to PLANES. Users were chosen who were already familiar with aviation, flight, and maintenance operations. Out of a total of 402 queries they entered, 275 were understood correctly by PLANES. Ten were rejected by PLANES as unintelligible and 117 produced errors; that is, PLANES did something incorrect. Of the 117 errors, 77 were due to simple omissions in the dictionary, grammar or query generator and so could be easily fixed, while the other 40 failed because of inadequacies in the formalisms of PLANES and would be rather difficult to fix. 22 of the difficult errors were due to the query generator, where the heuristics led PLANES to answer a question different from what the user had intended. For example, when asked "How many planes had NOR hours in during the time period?", it counted all the planes that had an NOR hours field instead of counting only planes that had greater than 0 NOR hours. Nonetheless, we are encouraged by the performance of PLANES and feel that with a moderate amount of further work it will prove to be a useful practical program.

7. Engineering and cognitive science

During the 70's the limitations of the engineering approaches of the 60's became evident and the importance of developing a cognitive science has become more generally appreciated. The reasons for this are summarized below.

(1) Even in simpler settings, language use was more complex and varied than was expected. For example, even with a question answering system we see not just requests but examples of speech acts, declaratives, reference to discourse entities, metaphorical and non-literal usages of language, and the need for understanding metaknowledge.

(2) We see a conflict between portability of natural language systems and the specialization of given systems. Systems that contain enough knowledge to be useful in a limited domain often are not very portable -- that is, great amounts of information must be added to them in order to be useful for another domain. Portable systems require large amounts of knowledge to be programmed into them to make the system useful in the new domain. We recognize now the need for a science of this process. Furthermore, we need to appreciate that as reported in the work on the "mythical man/month" by Brooks [1976] that if we have a project which requires a thousand man months, we canot solve that problem by putting a thousand men on the project for one month. When group size exceeds five or more people most of the time is spent on communication with other group members about what they're doing and very little time is actually spent in coding or developing systems. We need a greater understanding of the process of writing such systems in order to be able to rapidly produce large scale natural language understanding systems.

(3) The representations that we have developed to date are inadequate. They are imprecise. For example, the word "in" in the phrases "The crack in the wall" and "The man in the room" represent very different kinds of relationships between the words in the sentences. No adequate theory of representation of such words has been developed to date. Furthermore, the representations that we've dealt with to date are in many ways inappropriate. A great deal of effort has been concentrated on syntactic differences between sentences. A well known example is the following: "I saw the man on the hill with a telescope." Earlier discussions of this sentence pointed out that the phrase "with a telescope" could refer either to the man, that is, the man on the hill could have a telescope, or it could refer to me; I could see the man through the use of the telescope; and each of these would lead to different parse trees. However, we now realize that the most important distinctions here are not captured by the syntactic structures, but are really at the level of inferences, e.g. about where I am with respect to the man on the hill and where the telescope is with respect to both of us, as illustrated in Figure 5.

Figure 6 illustrates that by adding other sentences to give greater context the sentence is no longer ambiguous. The process of both being able to represent appropriately the different meanings of this sentence and being able to find the appropriate representation given to two or more sentences in context is still a outstanding problem.

(4) Finally, many types of language simply could not be dealt with at all. As examples of these types of language, we can list physical events and actions, descriptions of scenes, maps, paths, and instructions; real conversations, arguments, debates, discussions; language about emotions, accounts of inner experience; meta-descriptions and theories; situation assessment reports; poetic language, humor, irony, lies; etc.

3. Current Topics in Natural Language Understanding

At this point I would like to discuss several current topics and give some feeling for the kinds of research that are being done now - the kinds of questions that are being asked. The topics I would like to look at include speech acts; handling of novel language, especially metaphor; applied natural language understanding systems; the handling of natural language that refers to spacial events; and the inclusion of "common sense" in natural language understanding programs.

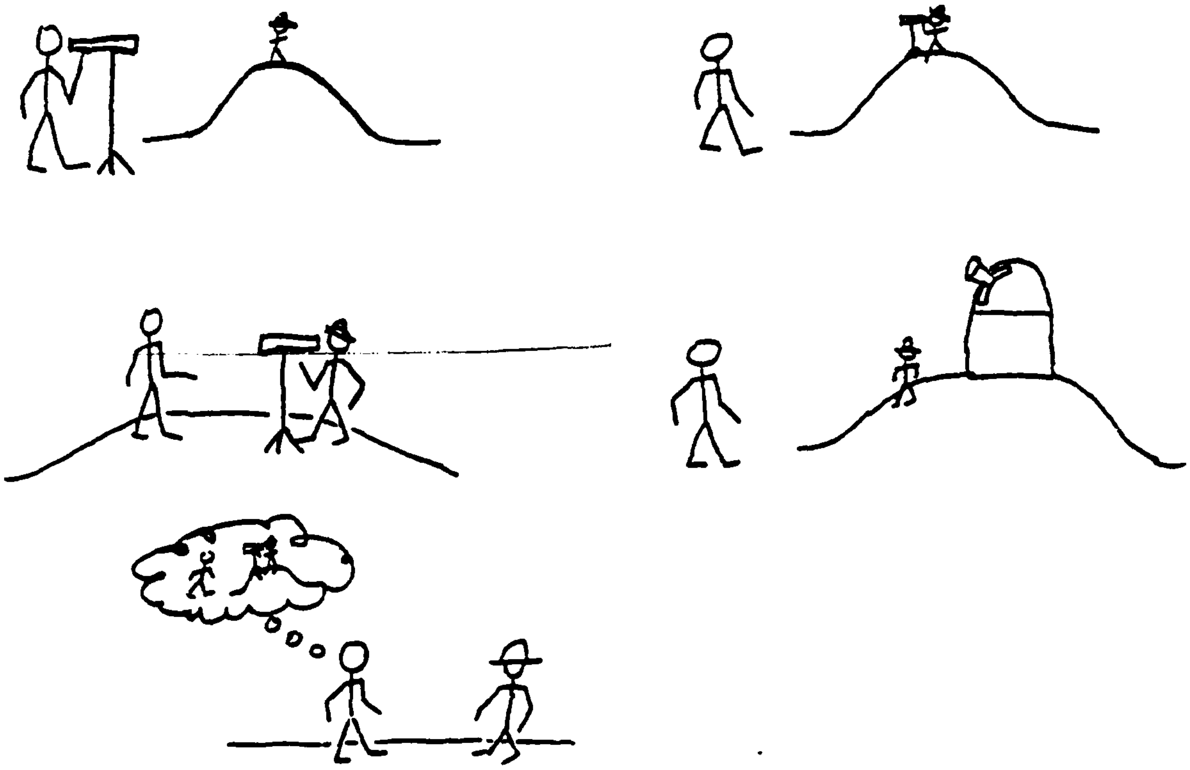

Figure 5. Depiction of various readings of "I saw the man on the hill with a telescope."

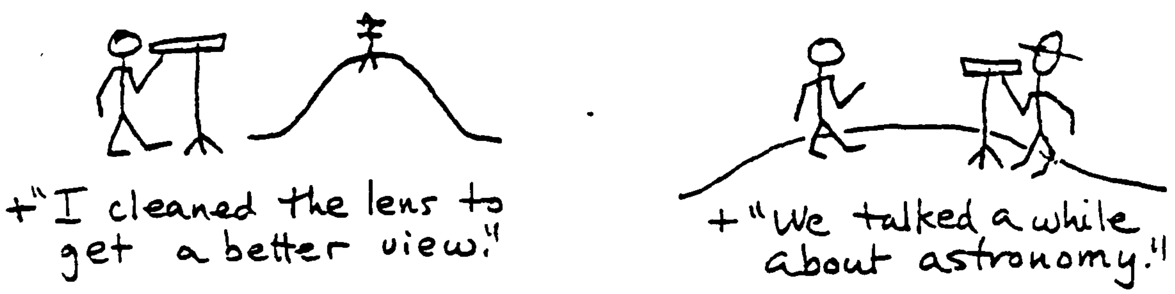

Figure 6. Adding another sentence to "I saw the man on the hill with a telescope." allows us to select a single reading.

3.1 Speech Acts

Speech acts are pervasive in the behavior of users toward natural language systems. As an example, consider this invented but possible dialogue:

User: Are there summaries for January?

System: Yes.

User: Could I have the January summaries?

System: Yes.

User: I would like the January summaries.

System: I understand.

User: Where are the January summaries?

System: I don't understand that.

User: Can you give me the January summaries?

System: Yes, I already told you that.

User: Would you please give me the summaries for January?

While this discourse seems rather ludicrous for a system with a wide range of possible behavior, it corresponds to a common problem with real systems, namely what Kaplan [1978] has called "stonewalling" behavior. One possible solution is to view each query as an instruction to select the closest matching procedure that the system has available to it. Such a solution is used in PLANES as well as in other current generation systems. A second solution which is only being investigated in a basic research context is to model the beliefs, intentions and goals of the user. While this second solution is much more difficult, it has a great theoretical interest and is probably the only solution which will ultimately result in a system that has a really satisfactory understanding of the requests of users. To give an idea of how difficult this second solution is, it is estimated that English contains over 1000 words each representing different kinds of speech acts. For example, agree, request, inform, state, demand, and imply are all different speech acts.

Another complicating factor in speech acts is that a single given speech act can serve many purposes. As an example, the sentence "That cake that you made looks delicious" might (1) inform the hearer of one's opinion about the cake: (2) inform the hearer of one's opinion about the cook's competence: (3) praise the cook or, (4) request some cake (or some more cake). As another example, the sentence, "No planes crashed in January, right?" both informs the hearer of the speaker's belief and asks for confirmation of the user's belief. In general, the understanding of speech acts is essential for any system which is to understand stories that contain dialogue, for understanding conversations, legal arguments, intelligence reports, political statements, as well as user input to natural language systems, and so on.

3.2 Novel Language

Even though a system may understand a large number of individual words, words can be used in combination to form concepts that are not easily understood as a simple combination of the sum of the parts. For example, consider the phrase "Water pump pulley adjustment screw threads damage report summary". A system may be capable of understanding each of the individual words and yet the overall phrase refers to a concept that is nowhere in the dictionary. Recent work by Finin [1980] and others has investigated how one could develop productive rules for generating the meaning of such long noun phrases. As another example, consider the sentence, "The tiger ran quickkly through the jungle." In order to understand the meaning of this sentence, one must realize that a tiger running quickly through the jungle will run at a different speed than a tiger running quickly across a plain. That is to say, the words, "ran quickly" are relative to the terrain through which the tiger is running. We do not have good facilities for handling language of this type. (By "handling", I mean being able to represent the different possible meanings differently, and being able to decide which meaning is intended in a given instance.)

Another important category of novel language includes metaphor, simile and analogy. During recent years we have come to understand much better how pervasive such phenomena are in language. Metaphor can be broadly divided into two types, which I will call "small" and "large". As an example of "small" metaphor, there is the sentence, "The thought escaped me like a squirrel darting behind a tree" [Ortony 1971]. In this example note that we have no way of talking about behavior of thought other than metaphorically. Even saying "the thought escaped me" is using a metaphor. Also note that in order to understand this we have to have some notion of the behavior of squirrels, the physical meaning of the rest of the metaphor.

As an example of a large metaphor, consider what I call the "hydraulic metaphor for economics." The hydraulic metaphor can govern large portions of a text, perhaps an entire book on economics. We need the hydraulic metaphor to understand terms like pressure, accumulation, cash flow, cash reservoirs, inflationary pressure, draining of resources", etc. As another example of a large metaphor, consider the "conduit metaphor" for communication, which treats language as though thoughts were capable of being put into a bundle and sent through a pipe to the hearer where they are unpackaged and inspected. Thus, we can say, "You aren't getting your ideas across to me" or "I gave her some good ideas". Other metaphors are possible for communication including the "radio link" metaphor as in, "We were on the same wavelength" or "I hear you loud and clear". It seems that some concepts can only be treated metaphorically and have no neutral terms in which to be expressed. As an example, love can be talked about as though it were a team effort, a physical connection or bond, a master-slave relationship, a resonance between two people, a journey, complementary shapes, sharing, fighting or contention, madness, etc.

For understanding novel language we can identify two extreme approaches. One is to have a number of canned concepts coupled with weak matching rules to select the most appropriate canned concept. This approach has been used to date for most natural language understanding systems. In general, however, we need other kinds of methods for reasoning about and ultimately producing new concept representations given old concept representations. Such methods have only begun to be explored.

3.3 Plausibility Judgement and Common Sense

The third current research topic I would like to discuss is the problem of modeling common sense and plausibility judgement in a language system. In general, natural language processing systems have had no connection with the perceptual world. That is to say, their only channel to the outside world has been through language. In a strict sense, such systems cannot be said to know what they are talking about, but can only know how to talk about things. They have no connection to the perceptual world, no more than rudimentary plausibility judgement, no ability to handle language about scenes, physical events and objects shapes, no good way for handling metaphors that attempt to interpret the abstract world in terms of the sensory-motor world, and no facility or even hope of a facility for doing realistic reasoning from experience (except linguistic experience).

Thus, current systems are unable to handle words such as attract, repel, divide, separate, connect, join, shatter, smash, scratch, cut, slice, crack, touch, hit, lean, support, hang, bounce, warp, wear, bend, tear, chip, crease, etc. Note that these words are extremely important not only in describing the physical world but also in describing abstract worlds, in particular relationships and interactions between people. Furthermore, no current systems can handle adverbial modifiers such as almost, violently, gently, hard, suddenly, fast, slow, etc.

Plausibility judgements are important even in the rather simple world of question answering systems for avoiding costly data base or memory search in the cases where questions are asked that simply make no sense. For example, if asked, "Did any plane crash more than five times last month?", a system should not have to go to the data base and search it in order to answer the question, since crashes usually happen to a given plane only once. It is possible that a user would expect a system to interpret "any plane" as referring to a class of aircraft, e.g. A7, F4, DC-10, or 747. Note that in order to make this interpretation of this sentence, however, a system would have to understand that "any plane" could not refer very meaningfully to a single aircraft, even though the language of the sentence would typically suggest that only a single aircraft was intended, as in "Did any plane have more than ten hours of maintenance last month?" In this latter case we would not want to interpret "any plane" as referring to DC-10 or F4 but rather as referring to a specific individual aircraft.

As another example of the need for plausibility judgement consider the following: "How many propellor replacements were made for A4's?" In this case a system that looked for propeller replacement examples when A4's had no propellers would waste resources in a serious manner. Here the problem is how to represent the equipment or nature of the items in the data base so that the data base search could be avoided. As another example, consider the following: "We were afraid the milk might make the baby sick, so we boiled it." Here, in order to to realize that "it" must refer to the milk and not to the baby, a system must understand the ordinary behavior of people, in particular that people might boil milk but would rarely boil babies.

We have recently made progress in dealing with simple aspects of space. Since space is a topic that can be shared by many possible natural language system applications, many possible domains, we feel it is useful to consider it in the abstract.

Programs written by Lois Boggess [1978][Waltz and Boggess 1979] can deal with the following types of sequences of input: The goldfish is in a goldfish bowl; the goldfish bowl is on a shelf; the shelf is on the desk; the desk is in a room. If given a question, "Is the goldfish in the room?" the system can answer "yes" by referring to a representation that it builds as shown in Figure 7.

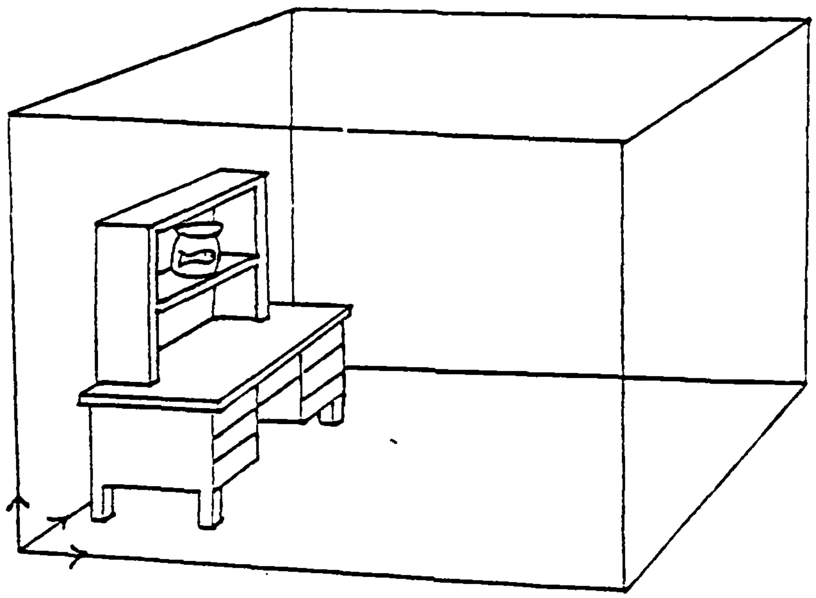

Figure 7

The system can simply check to see that the goldfish's coordinates lie within the intervals defined by the corners of the room and answer "yes" easily. Previous approaches to answering such questions involved processes similar to those used in theorem proving, where one would try to apply a theorem, for example, that if a is in b, and if b is in c, then a is in c, and then answer the question, "Is the goldfish in the room?" by establishing the validity of a sequence of theorem-like steps. In general, there may be many theorems, since words such as "in" can have many meanings: compare the crack in the wall, the desk in the room, the water in the glass, the face in the mirror. There will in general be different theorems for each of the possible meanings of each word, so that answering questions may quickly come to involve a large amount of search. In this example, we can see that dealing with space, metrics, coordinate systems, and distances provides a much simpler and more direct solution to this kind of problem.

As an example of the usefulness of such ability to deal with space, we can now see easily how to handle the question which caused Bar-Hillel to quit the field of natural language processing in the early 60's. The sentence that Bar-Hillel cited as impossible to solve by machine systems was: "The box is in the pen". In order to understand this statement, we have to realize that "pen" cannot refer to an ordinary writing implement, which is rather fixed in size and smaller than most boxes. The pen, in this case, more likely refers to a play pen or a stock pen. Using a program similar to Boggess's, mentioned above, we can note that boxes have a size range from perhaps two inches on a side to five feet on a side, that they are hollow containers, and that pens -- that is, writing implements -- are on the order of six inches long and a half-inch in diameter and relatively fixed in size and not hollow containers; that pens -- play pens -- are roughly four feet on a side and hollow containers. Stock pens are even larger. When a system tries to construct a special representation of a box within a pen (writing implement) the system is simply incapable of creating such a relationship. On the other hand, in the case of a box within a playpen or stock pen, the representation could be easily constructed and the system can thus plausibly judge that "pen" in this case must refer to stock pen or playpen.

3.4 Progress

In this section I would like to point out, for the interested reader, a number of recent projects that promise to solve some of the outstanding problems of natural language understanding.

The first topic of interest is knowledge representation. A number of pieces of work in recent years have led to considerable progress in the knowledge representation area. The KL-ONE system of Brachman [1979] is general enough to represent grammar rules, semantic interpretation rules, speech act rules, as well as object and event taxonomies. KL-ONE is a language in which many different kinds of knowledge can be expressed in a uniform manner, and shared between different components of a full natural language understanding system. KL-ONE was inspired by the "procedural semantics" ideas of Woods [1979].

Other work on representing mechanisms and the geometry of objects has been done by Rieger [1975], Hayes [1978], Forbus [1979], deKleer [1977, 1979], and Waltz [1979]. Work in representing physical scenes and events has been done by Boggess [1978], Waltz [1980a,b], Herskovitz [1980], and Johnson-Laird [1979]. Representation of large scale space maps has been explored by Kuipers [1978] and McDermott [1980].

Other work in knowledge representation has attempted to deal with problems of inconsistent knowledge -- that is, the problem of how to add new information to a system which may conflict with information presently in a system. Work on "non-monotonic logic" to deal with such conflicts has been done by Doyle [1978, 1980], McDermott [1979], Weyrauch [1979] and others.

Some other examples of general progress are in the area of summarizing and translating newspaper articles, modeling emotional conflicts and reactions, modeling argumentation, writing psychologically realistic parsers, making simple natural language front ends commercially available, and understanding the meanings of phrases. Recently there has been a great deal of interest in the generation of language and also renewed interest in the area of speech understanding. Furthermore, many natural language parsers are approaching closure, that is, being able to handle all naturally occurring grammatical types of sentences (see, for example, [Bobrow 1980]).

9. Summary

I have tried to show in this paper how ideas have progressed from the point where we first understood that computers could be used for processing text and general concepts as well as numbers to the point where simple mechanisms for dealing with language were tried but discarded for machine translation, through an era of attempts to handle natural language processing through any means available, through the engineering of systems that deal with simplified natural language in narrow domains. We are now at a phase where we have begun to realize that in order to deal with natural language, we have to understand better how it is that people process language, so our emphasis has shifted from engineering to cognitive science. If we are to have natural language understanding systems that are truly satisfactory, it must be the case that natural language systems make appropriate inferences about the natural language of people. It must also be the case that if a computer system presents English output to users, the user is justified in making the inferences one would ordinarily make, given that language. In order to be able to meet these two criteria, natural language systems must not simply understand the shallow surface meaning of language, but must also be able to understand the deeper implications and inferences that a user is likely to intend and likely to take from language. In order to do this, the systems must be capable of understanding user goals, intents, and strategies, as well as multiple purposes served by any given piece of language.

Secondly, we have come to understand much more clearly that if we are to ever build natural language systems with both depth and breadth, we must come to grips with either the problem of learning from experience, or the problem of designing and building software systems of a scope and subtlety beyond anything yet accomplished. In either case, we lack the knowledge of how to proceed. It seems arrogant to assume that we could program a natural language system to reach adult competence in language in anything less than the twenty years required by humans, and, as argued by Brooks, we cannot simply accomplish this task by putting more and more people on the same project. Not only is it difficult to coerce such people to work effectively as a team because of the sheer amount of intercommunication necessary; such an approach to writing large natural language processors also requires that we understand all the knowledge representation schemes ahead of time so that all team members can generate code portions that will work properly together. At the present such a massive approach to a natural language understanding system is simply not feasible, and there seems to be no prospect for anything other than narrow domain natural language systems for the foreseeable future. Areas that still need a great deal of work include representation of space, time, events, human behavior, emotions, physical mechanisms, and many processes associated with metaphor. Furthermore, we must face the problems associated with learning from experience. Even if we are able to program a system which has adult competence in language, such a system, if it is to display language processing behavior like an adult, must also be capable of learning and dealing with new concepts that are taught to it by a user or through experience. We as yet have very few ideas on how to deal with such phenomena.

Finally, at this point in history there are many opportunities. We have some natural language systems which are already useful and a number of others which should be usefully applied within the near future. We also have seen continued dramatic improvements and increases in the power of available hardware and software. Each advance brings real time natural language processing closer and closer. Natural language systems have for their entire history pushed the limits of available computation, and increases in the computational power available to users will clearly aid in the solution of natural language processing problems. Most computers have not been designed to work well with natural language processing systems. Computers have been tuned primarily for numerical problems. With the wide availability of VLSI technology it will be possible for natural language processing researchers to specify and obtain computers which have architectures appropriate to the natural language processing tasks. Some natural candidates for improvements in this area include true associative memories, memories with highly distributed processing, separate processors for syntactic, semantic, pragmatic, and processing phases of language, hardware for analyzing speech, and so on. Already work is underway in these areas.

BIBLIOGRAPHY

Becker, J. D. 1975. The phrasal lexicon. BBN Rpt. No. 3081, Bolt Beranek and Newman, Inc., Cambridge, MA.

Bobrow, D. G. 1968. Natural language input for a computer problem-solving system. In Semantic Information Processing, M. L. Minsky (ed.), MIT Press, Cambridge, MA, 146-226.

Bobrow, R. J. and B. L. Webber. 1980. Knowledge representation for syntactic/semantic processing. Proc. 1st Annual Natl. Conf. on Artificial Intelligence, August, Stanford, 316-23.

Boggess, L. C. 1978. Computational Interpretation of English Spatial Prepositions. Tech. Rpt. T-75, Coordinated Science Lab, Univ. of Illinois, Feb. 1979.

Brooks, F. P. 1975. The Mythical Man-month. Addison-Wesley, Reading, MA.

Colby, K. M., B. Faught, and R. Parkinson. 1974. Pattern matching rules of the recognition of natural language dialogue expressions. Stanford AI Lab, Memo AIM-234, June 1974.

de Kleer, J. 1977. Multiple Representations of Knowledge in a Mechanics Problem-Solver. Proc. 5th Int'l. Joint Conf. on Artificial Intelligence. Cambridge, MA: MIT, pp. 299-304.

de Kleer, J. 1979. The Origin and Resolution of Ambiguities in Casual Arguments. Proc. IJCAI-79. Tokyo, Japan, pp. 197-203.

Finin, T. W. 1980. The semantic interpretation of nominal compounds. Tech. Rpt. T-96, Coordinated Science Lab, Univ. of Illinois, Urbana, March 1980.

Forbus, K. D. 1980. A Study of Qualitative and Geometric Knowledge in Reasoning about Motion. M.S. Thesis, Massachusetts Institute of Technology (February).

Green, B. F., et al. 1963. Baseball: An automatic question answerer. In Computers and Thought, Feigenbaum and Feldman (eds.), McGraw-Hill, New York, 207-33.

Grice, H. P. 1975. Logic and conversation. In Syntax and Semantics: Speech Acts, P. Cole and J. L. Morgan (eds.), Academic Press, New York, 41-58.

Harris, L. 1977. User oriented data base query with the ROBOT natural language query system. Intl. J. of Man-machine Studies 9, 697-713.

Hayes, P. J. 1978. The Naive Physics Manifesto. Unpublished paper (May).

Heidorn, G. E. 1974. English as a very high level language for simulation programming. Sigplan Notices 9, 91.

Hendrix, G. G., E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. 1978. Developing a natural language interface to complex data. ACM Transactions on Database Systems. Vol. 3, No. 2 (June).

Herskovitz, A. 1980. On the Spatial Uses of Prepositions. Proc. of 18th Annual Meeting of the Association for Computational Linguistics, Univ. of Pennsylvania, Philadelphia, June 19-22.

Johnson-Laird, P. N. 1980. Mental models in cognitive science. Cognitive Science, Vol. 4, No. 1, pp. 71-115.

Johnson-Laird, P. N. 1980. Mental models of meaning. To appear in Joshi, Sag and Webber (eds.), Elements of Discourse Understanding, Cambridge University Press, 1980.

Kaplan, S. J. Cooperative responses from a portable natural language data base query system. (Univ. of Pennsylvania Ph.D. dissertation) Tech. Rpt. HPP-79-19, Computer Science Dept., Stanford Univ., July 1979.

Kuipers, B. J. 1977. Representing Knowledge of Large-Scale Space. Massachusetts Institute of Technology Artificial Intelligence Laboratory, Report No. AI-TR-418 (July).

Lehnert, W. G. 1977. A conceptual theory of question answering. Proc. 5th Intl. Joint Conf. on Artificial Intelligence, Vol. 1, August, MIT, Cambridge, MA, 158-64.

Malhotra, A. 1975. Design criteria for a knowledge-based English language system for management: An experimental analysis. MAC TR-146, MIT. Cambridge, MA.

McCarthy, J. 1968. Programs with common sense. In Semantic Information Processing, M. L. Minsky (ed.), MIT Press, Cambridge, MA, 403-418.

McDermott, D. 1980. Spatial Inferences with Ground, Metric Formulas on Simple Objects. Yale University, Dept. of Computer Science Research Rept. 173 (January).

Minsky, M. L. 1968. (ed.) Semantic Information Processing. MIT Press. Cambridge, MA.

Minsky, M. A. 1975. A framework for representing knowledge. In The Psychology of Computer Vision, P. Winston (ed.), McGraw-Hill, New York, 211-77.

Plath, W. J. 1976. Request: A natural language question-answering system. IBM J. Res. Devel. 20, 4, 326-35.

Rieger, C. 1975. The commonsense algorithm as a basis for computer models of human memory, inferene, belief and contextual language comprehension. In R. Schank and B. Nash-Webber (eds.), Theoretical Issues in Natural Language Processing, ACL, Arlington, VA, 180-195.

Schank, R. C., et al. 1973. Margie: Memory, Analysis, Response Generation, and Inference on English. Proc. 3rd Intern. Joint Conf. on Artificial Intelligence, August, Stanford, 255-61.

Schank, R. and R. Abelson. 1975. Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Assoc. Hillsdale, NJ.

Searle, J. R. 1970. Speech Acts. Cambridge University Press. Cambridge, England.

Tennant, H. R. 1980. Evaluation of natural language processors. Tech. Rpt. T-103, Coordinated Science Lab. Univ. of Illinois, Urbana, Dec. 1980.

Thompson, F. B., et al. 1969. REL: A rapidly extensible language system. Proc. 24th Nat'l. Conf. ACM, p. 399. New York, NY.

Thompson, F. B. and B. H. Thompson. 1975. Practical natural language processing: The REL System as prototype. In Advances in Computers, 13, p. 109, M. Rubinoff and M. C. Yovtis (eds.). Academic Press. New York, NY.

Waltz, D. L. 1978. An English language question answering system for a large relational database. Comm. ACM 21, 7, 526-39, (July).

Waltz, D. L. 1979. Relating images, concepts, and words. Proc. of the NSF Workshop on the Representation of 3-D Objects, University of Pennsylvania, Philadelphia.

Waltz, D. L. and L. Boggess. 1979. Visual analog representations for natural language understanding. Proc. 6th Intl. Joint Conf. on Artificial Intelligence, Vol. 2, August, Tokyo, Japan, 926-34.

Waltz, D. L. 1980a. Understanding scene descriptions as event simulations. Proc. 18th Annual Meeting of the Association for Computational Linguistics, June, Univ. of Pennsylvania, Philadelphia, 7-12.

Waltz, D. L. 1980b. Generating and understanding scene descriptions. In Joshi, Sag, and Webber (eds.), Elements of Discourse Understanding, Cambridge University Press, to appear; also Working Paper 24, Coordinated Science Laboratory, Univ. of Illinois, Urbana (Feb. 1980).

Weizenbaum. 1966. ELIZA - A computer program for the study of natural language communication between man and machine. Comm. ACM 10, 8, 474-80.

Wilks, Y. A. 1975. A preferential, pattern-seeking, semantics for natural language inference. Artificial Intelligence, 6, 1, 53-74.

Winograd. 1972. Understanding Natural Language. Academic Press. New York, NY.

Woods, W. A., R. M. Kaplan, and B. Nash-Webber. 1972. The lunar sciences natural language information system: Final report. BBN Report 2378, Bolt Beranek and Newman, Inc. Cambridge, MA.

Woods, W. A. 1970. Transition network grammars for natural language analysis. Comm. ACM 13, 10, 591-606.

DATE
ILMED
-8